Malaysian Journal of Computing, 7 (1): 1036-1046, 2022

eISSN: 2600-8238 online

AN IMPROVEMENT OF RESOURCE CONSUMPTION IN WIRELESS SENSOR NETWORK (WSN) USING COMPRESSIVE SENSING

Maizatul Akmal Ibrahim[1*] and Norkhushaini Awang[2]
[1,2]*Faculty of Computer and Mathematical Sciences, Universiti Teknologi MARA, Shah Alam, Malaysia*
[1*]akml.ibrhm@gmail.com, [2]shaini@tmsk.uitm.edu.my

ABSTRACT

Wireless Sensor Network (WSN) refers to a group of spatially dispersed and dedicated sensors designed to monitor and record the physical conditions of the environment and organise the data collected at a central location. One of the problems in wireless sensor network (WSN) is the low computational resources available which is the limited battery life of sensor nodes. Hence, one of the ways to prolong the lifetime is to implement Low energy Adaptive Clustering Hierarchy (LEACH). The Compressive Sensing (CS) algorithm has been implemented to the network through Cluster Head (CH) to further the lifetime longevity. CS is a technique utilised for energy efficient data gathering in WSNs. Compressive sensing provides benefits such as robustness, long lifetime of the network, reduced energy consumption, and a simple routing scheme. The simulation testing and algorithm implementation have been done in MATLAB with the scripting language. Through the implementation of CS, it can be concluded that the CS algorithm helps increase the network lifetime by 9.7%.

Keywords: *Compressive Sensing, Energy Consumption, LEACH, WSN*

Received for review: 22-10-2021; Accepted: 22-03-2022; Published: 31-03-2022

1. Introduction

Wireless Sensor Network (WSN) refers to a group of spatially dispersed and dedicated sensors designed to monitor and record the physical conditions of the environment and organise the data collected at a central location. WSNs are an evolving technique used to gather data from several environmental phenomena. It has a range of diversified fields of use, including home, workplace, automation and control, transport, health care, monitoring of the environment, protection and surveillance, and monitoring and detection of vehicles (Rehmani & Pathan, 2016). Sensor nodes require more resources for connectivity than any other for communication. If there is an issue of power exhaustion, they often become invalid, and there is no way to recharge them (Hu, 2012). Given the situation that the sensor nodes in WSN have a non-renewable energy supply, compressive sensing (CS) is beneficial to be implemented as it has the advantage of reducing energy consumption and prolonging network lifetime. Compressive sensing is a signal processing method for the efficient acquisition and reconstruction of a signal by finding solutions to undetermined linear systems. This method is based on the principle that a signal's sparsity can be exploited through optimisation to recover it from far fewer samples than the Nyquist-Shannon sampling theorem requires. The compressive sensing in WSN allows for a reduction in the amount of data that must be transferred and stored and the recovery information from any requesting node (Martinez *et al.*, 2016). Therefore, this project proposes the implementation of a compressive sensing algorithm into WSN to reduce the energy consumption of the data transmission.

2. Literature Review

A wireless sensor network (WSN) is an interconnected wireless network consisting of independently configured devices, known as nodes, that use sensors to monitor the conditions of their environment (Olakanmi & Dada, 2018). The WSN has many applications ranging from environmental monitoring, surveillance tasks, controlling machinery, target tracking, military defence, healthcare, and industrial monitoring. The position of sensor nodes is not required to be engineered or predetermined, allowing random deployment in inaccessible terrain or disaster relief operations (Garg, 2007). The wireless network consists of a mobile network, a wireless LAN, a Bluetooth network, an ad hoc network. The sensor network has many characteristics, such as mobility, switching character and battery power limit capability, similar to the Ad hoc network (Liu *et al.*, 2009).

2.1 Routing Protocols

Figure 1. Routing Protocols.

A few routing protocols are used in WSN, as illustrated in Figure 1. The first protocol is the data-centric routing protocol, and it integrates data from several sensor nodes along a specific path. This protocol removes redundancy and reduces the overall quantity of data transmitted before transferring it to the base station. Direct diffusion, rumour routing, and the sensor protocol for information via negotiation (SPIN) protocol are examples of data-centric routing protocols (Olakanmi & Dada, 2018). The SPIN protocol is intended to compensate for flaws in other protocols, such as flooding and gossiping. The main idea is that, in contrast to meta-data, which is simply a description of the data sensed by the node, sharing data sensed by the node may need more resources (Shabbir & Hassan, 2016). Following is the hierarchical routing protocol. It groups network nodes into hierarchical clusters. The protocol selects a node with high residual energy to serve as the cluster head for each cluster. Each cluster node's sensed data is sent through the network clusters' cluster heads (Olakanmi & Dada, 2018). Low energy adaptive clustering hierarchy (LEACH) is a routing technique that organises the cluster so that all sensor nodes in the network divide energy evenly. The LEACH protocol creates many clusters of sensor nodes and a node designated as the cluster head, which acts as the routing node for all of the other nodes in the cluster (Shabbir & Hassan, 2016). The location-based routing protocol routes information based on source and destination node distances. It calculates the distance between source and destination nodes to determine estimated routing energy. Location information allows the network to choose the best route (Olakanmi & Dada, 2018). An example of location-based protocols is Geographic Adaptive Fidelity (GAF). GAF is a location-based energy-sensitive routing algorithm primarily designed for mobile ad hoc networks and used in sensor networks. This protocol aims to optimise the performance of wireless sensor networks by identifying equivalent nodes with packet forwarding (Lee *et al.,* 2014).

The QoS-based routing protocol balances efficient data delivery to the sink node with some predefined QoS metrics. Some of the existing QoS-based routing protocols are SPEED and QoS-aware and heterogeneously clustered routing (QHCR) (Olakanmi & Dada, 2018). Each SPEED node broadcasts a beacon packet to its neighbours periodically. This periodic beaconing is only used between neighbours to exchange information (Kordafshari *et al.*, 2009). For real-time applications and delay-sensitive applications at lower energy, the QHCR protocol provides dedicated paths. The QHCR protocol consists of collecting information, selecting cluster heads, and stages of intra-cluster communication (Olakanmi & Dada, 2018). The mobility-based routing protocol is a simple protocol that assures data transportation from source to destination nodes. Some of the examples are Tree-based efficient data dissemination protocol (TEDD) and scalable energy-efficient asynchronous dissemination (SEAD) (Olakanmi & Dada, 2018). TEDD is a technique for distributing data in an energy-efficient manner using a mobile sink. Initially, it creates a tree with a root node (Sharma & Jena, 2014) and SEAD also consider the distance and packet traffic rate between nodes while generating near-optimal dissemination trees. Sinks can move while still receiving data updates if they do not communicate their location to the tree (Kim *et al.*, 2003).

2.2 Low Energy Adaptive Clustering Hierarchy (LEACH)

Low Energy Adaptive Clustering Hierarchy (LEACH) routing protocol is classified into the hierarchical routing protocol, and it is one of the energy-efficient protocols. It’s also a self-organising, adaptive clustering protocol that distributes the energy load evenly throughout the network’s sensor nodes using randomisation-based probability (Singh, 2015). The network is divided into different clusters, each with its own Cluster Head (CH), which is connected to the cluster members nodes, and a Base Station (BS) that collects the aggregated data from the cluster head. Because the cluster head has more functions than the other nodes, it consumes energy more quickly than the others, causing it to expire sooner (Nasr & Quwaider, 2020).

The disadvantage of the LEACH routing protocol is that it does not take the residual energy of nodes into account when selecting cluster head nodes; the selected node must act as the cluster head node regardless of residual energy. Furthermore, the cluster head node is chosen at random, which cannot ensure that the cluster head is distributed evenly, and the different number of cluster member nodes will result in different energy consumption of the cluster head, and the algorithm assumes all nodes have the same initial energy (Yufeng *et al.*, 2013).

2.3 Compressive Sensing

Compressive sensing (CS) is a signal processing technique used to efficiently acquire and reconstruct a signal from a reduced set of measurements by using prior knowledge of the signal’s sparsity on some basis. The Nyquist–Shannon sampling rate efficiently samples the data at a rate significantly lower than the classic requirement in uniform sampling (Khosravy *et al.*, 2020). In short, a lower number of measurements produced by multiplying a sufficiently large signal by a well-defined matrix can be retrieved with a high probability. (Taghouti, 2020).

In data collection over WSNs, compressive sensing becomes important as it can significantly improve the efficiency of the routing protocol. The main goal of applying CS to WSN data aggregation is to reduce the cost of communication. CS is used to significantly reduce the number of transmissions in WSNs by working together with network coding (Ifzarne *et al.,* 2020). CS is a concept originating from the signal processing field. The strength of compressive sensing is its ability to reconstruct a small number of measurements from a sparse or compressible signal without requiring any prior knowledge of the structure of the signal. CS is beneficial when signals are sparse on a known basis, measurements are costly, and computations at the receiver end are inexpensive (Masoum *et al.*, 2013). CS includes different ways to represent a signal with a limited number of measurements and reconstruct it from those measurements. The effectiveness with which the original signal has been retrieved from the compressed information is critical in the compressive sensing framework. Compressive sensing reconstruction can be divided into a few classifications, namely convex relaxation, non convex relaxation algorithm, iterative threshold algorithm, Bregman iterative algorithm, and combinatorial algorithm, as shown in Figure 2.

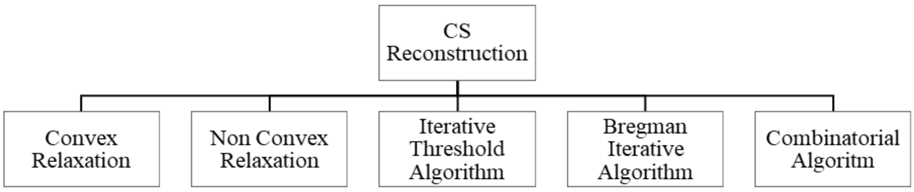


Figure 2. Compressive Sensing Reconstruction.

Convex relaxation is the process of taking a non-convex problem that has to be solved and replacing it with a convex problem that can be solved. Typically, this is accomplished by either convexifying the constraints or convexifying the functional constraints. (Romberg, 2015). To get reconstruction, it solves a convex optimisation problem using linear programming. Although the number of measurements needed for perfect reconstruction is low, the approaches are computationally demanding. (Luiz et al., 2016).

There are many important practical problems that are non-convex, and most non-convex problems are challenging to solve precisely in a reasonable period. As a result, the idea of employing heuristic algorithms, which may or may not provide desired results, was developed. The optimisation is carried out through different minimisation approaches with certain variables maintained constant in a cyclical way and linearisation techniques, in which the objectives and constraints are linearised or approximated by a convex function (Luiz et al., 2016). For the iterative threshold algorithm, Blumensath and Davies (2010) proposed the Iterative Hard Thresholding (IHT) method for recovery in a compressed Sensing scenario, and implementation of this approach has demonstrated its theoretical guarantee. The main idea behind IHT is to look for a good candidate for an estimate of support set that fits the measurement (Luiz et al., 2016).

The Bregman iterative algorithms have given an efficient and straightforward method for addressing the l1 minimisation issue using the Bregman iterative algorithm (Luiz et al., 2016). It presents a new approach that provides accurate solutions to restricted issues by addressing a succession of unconstrained sub-problems created by a Bregman iterative regularisation scheme (Manchanda & Sharma, 2020). The combinatorial algorithm is a class of group testing techniques to recover sparse signals. Compared to convex relaxation or greedy algorithms, it is highly fast and efficient (Luiz et al., 2016). For Φ to be sparse, a specific prototype in the measurements is required for this algorithm (Manchanda & Sharma, 2020).

3. Methodology

This chapter is about analysing the result obtained from the testing phase that has been conducted. Both LEACH and compressive sensing are simulated using MATLAB version R2020a. The project managed to display the information needed for performance evaluation, such as tested and reconstructed image details, the energy consumption of nodes per 100 rounds, the number of alive nodes per 100 rounds and the first dead node of both simulations of WSN.

3.1 Implementation of Wireless Sensor Network

Compressive sensing has been tested using an image before implementing it on WSN. Figure 3 shows the original image, simplifying the simulation testing and cutting down the simulation runtime, and the image has been downsized by 40%. The information of the image is as stated in Table 1.

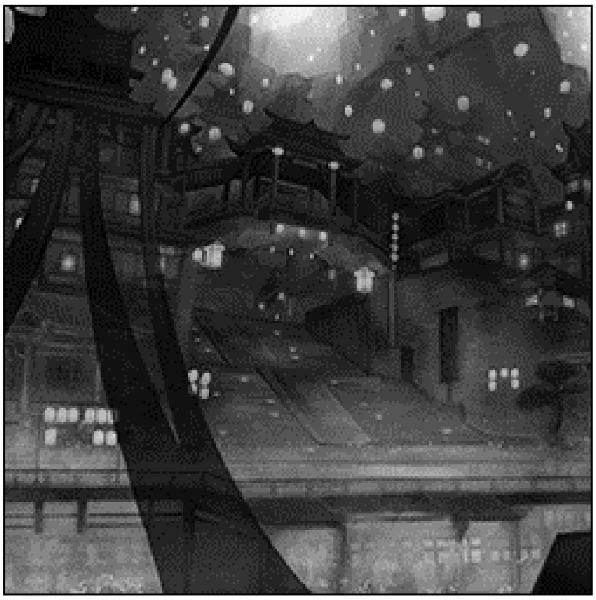


Figure 3. Original Image.

Table 1: Details of the Original Image.

	Original Image	Resized Image
Dimensions (pixel)	250x250	150x150
Size (bytes)	111,385	47,022
Resolution (pixel)	72	72

3.2 LEACH Simulation

In this subtopic, an explanation of LEACH simulation using MATLAB has been laid out. Figure 4 is the random generated WSN Topology with the details of the simulation environment setup as in Table 2. In LEACH, the initial data packet size of 4000 bytes is affected when compressive sensing is implemented. The simulation testing has made a few assumptions. Firstly, all nodes in the network have the same initial energy. Next, throughout the simulation, all nodes are static, and data is always sent from nodes.

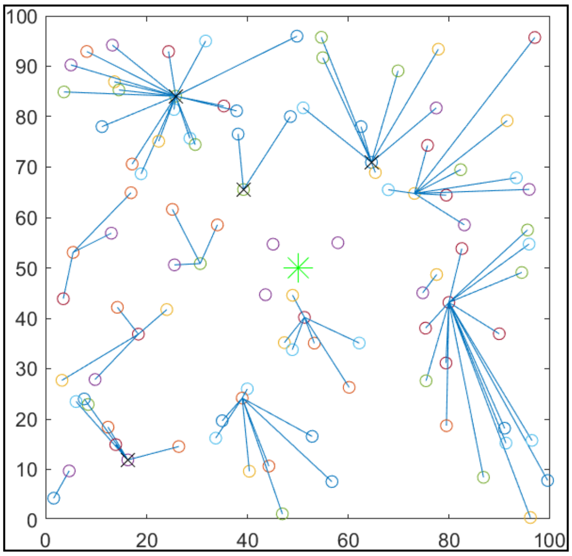


Figure 4. WSN Topology at the Beginning of Simulations.

Table 2. LEACH Parameters of Simulation.

Parameter	Value
Sensor deployment area	100 x 100 m
Base Station Location	(50,50) m
Number of nodes	100
Energy Model	Battery
Data Aggregation Energy	5*0.000000001 J
Initial energy of sensor	0.5 J
Probability of node to become cluster head	0.1
Maximum round	5000

Simulations were conducted against a sensor area of 100x100 meters, and a sink node was set in the middle of the area. Also, 100 battery-powered nodes have been deployed in the sensor area. Next, the initial energy of each node is set to 0.5 joules with the data aggregation energy of 5*0.000000001 joules. Furthermore, the probability that the node would act as the cluster head at a time has been set to 0.1 probability, and lastly, the maximum lifetime of the simulation is 5000 rounds. With the parameter has been stated, the following topic is the result of the simulation.

4. Result and Finding

From the explanation that have been discussed, this study has conducted experiments to prove the use of compressive sensing. The code in Figure 5 has been used to apply the algorithm. It is also to obtain the information of M and N for the compression formula (1),

$$y = \Phi x \tag{1}$$

where Φ is MxN CS matrix, hence, $M = 1000$ and $N = 3600$ satisfy the condition that N “M.

```
A = A([40:99],[40:99]);
x = double(A(:));
n = length(x);
m = 1000;
```

Figure 5. Compression Sensing Formulation.

The result of the reconstructed compressive sensing algorithm is shown in Figure 6 with the least-square problem and basis pursuit minimisation. The results were obtained through 60x60 pixels of the resized images and have been classified into variable A.

```
if (strcmp(PacketType,'Hello'))
      PacketSize=Model.HpacketLen;
else
      if (strcmp(PacketType,'Data-CH'))
            PacketSize=Model.DpacketLen*0.14;
      else
            PacketSize=Model.DpacketLen;
      end
end
```

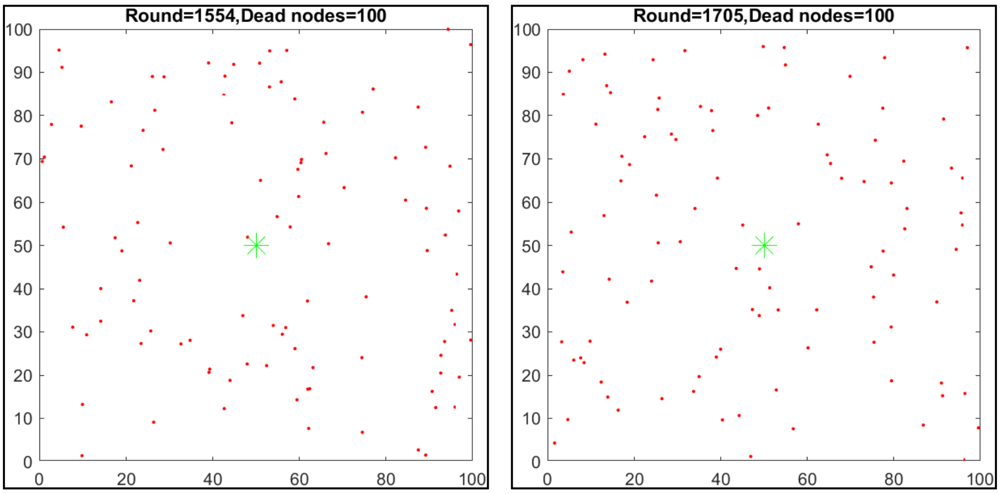
Figure 6. Packet Size Calculation.

Table 3 shows the information of the tested image. The tested image only took 40% of the resized image for testing with the size of 2,686 bytes, and it is done to reduce the processing time. According to the table, the size has been reduced to 14%, from 2,686 bytes to 376 bytes. This reduction rate is next implemented in LEACH simulation, specifically, on Cluster Head.

Table 3. Details of Tested Image.

	Tested Image	**Compressed Image**	**Reconstructed Image**
Dimensions (pixel)	60x60	60x60	60x60
Size (bytes)	2686	376	1005
Resolution (pixel)	72	72	72

Compressive sensing is only implemented into the cluster head because it consumes more energy than other nodes since all nodes send their data to their cluster head. Hence, implementing compressive sensing to cluster head will prolong the lifetime of the cluster head and the network. The code in Figure 5 is how the rate is implemented into the cluster head in the LEACH simulation. The reconstruction rate of the implemented compressive sensing algorithm can be calculated to 37.42%, referring to Table 3.



(a) (b)

Figure 7. WSN Topology at the end of the simulation (a) without CS implementation and (b) with CS implementation.

Figure 7 shows the end of the simulation with both implementations. The green dot represents the sink, while the red dots are the dead nodes. While both show that all nodes have runs out of energy, what differs is how long the networks last, and it can be measured by the number of rounds that both networks manage to last. With the implementation of compressive sensing on cluster head, it has prolonged the network lifetime from 1554 rounds to 1705 rounds which is equivalent to an increment of 9.7%,

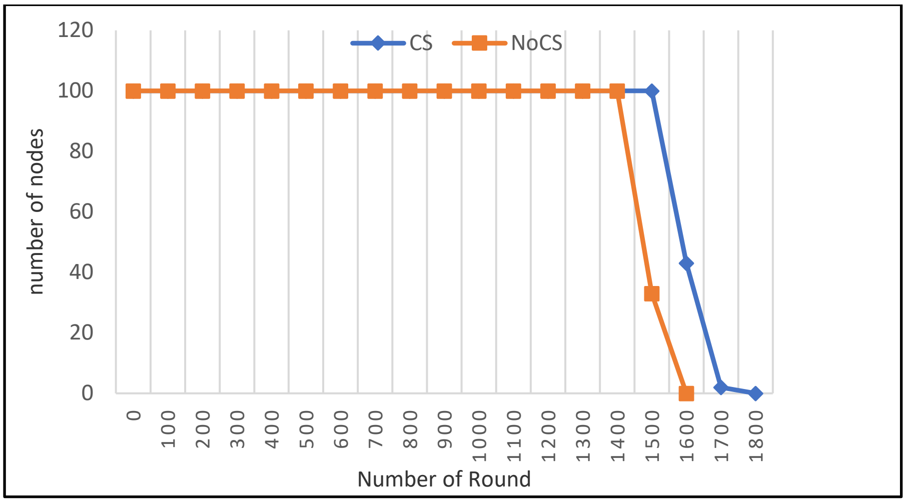


Figure 8. Number of Alive Nodes.

Table 4. First Dead Node.

Without Compressive Sensing	With Compressive Sensing
1409	1509

As shown in Figure 8, the number of alive nodes at the beginning of both simulations are 100. With the difference of 100 rounds, the simulation without compressive sensing implementation started having a dead node in round 1400, and the simulation with compressive sensing implementation started having a dead node in round 1500, as shown in Table 4.

In simulations conducted using an environment without compression sensing, at round 1500, the number of nodes decreased compared to the previous round by 77%. The 33 nodes and the network is entirely dead in round 1600 nodes. In the simulation with CS implementation, the performance is slightly better with the decrement of 57% in round 1600 from the previous round and in the next round, it went through 95% decrement to 2 nodes of 43 nodes from round 1600.

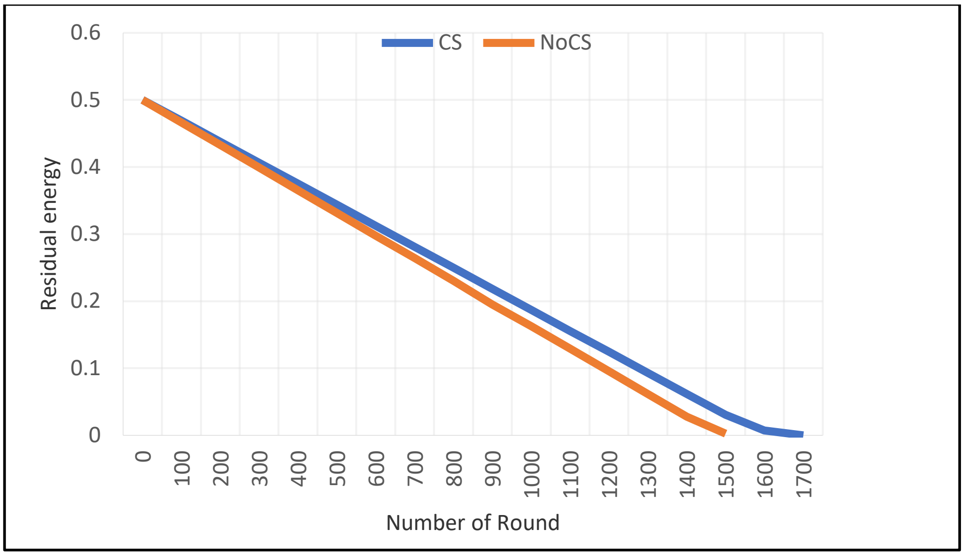


Figure 9. Residual Energy of Nodes.

Figure 9 shows the residual energy left in nodes through the rounds until all nodes are dead. The residual energy can observe that the network with compressive sensing implementation lasts slightly longer than the network without compressive sensing implementation. That happened due to the data compression applied to the packet in one of the environments. Hence, by implementing this simulation, it has been proven that compressive sensing can help in prolonging the network lifetime.

5. Conclusion

Wireless sensor network (WSN) is a large-scale wireless network that is infrastructure-less, and batteries typically power the nodes. The energy efficiency in this wireless network is a challenge due to the limited battery capacity of the nodes. Hence, this project proves that implementing a compressive sensing algorithm in the data before the transmission increased the network's longevity. Therefore, from the testing conducted, it can be concluded that the implementation of compressive sensing does increase the lifetime of the nodes. So, this leads to the increment of the network's longevity by 9.7%, and the number of rounds can observe it until all nodes are dead.

References

Blumensath, T., & Davies, M. E. (2010). Normalised iterative hard thresholding: Guaranteed stability and performance. *IEEE Journal on Selected Topics in Signal Processing*, 4(2), 298–309. https://doi.org/10.1109/JSTSP.2010.2042411.

Garg, V. (2007). Wireless Communications & Networking. In *Wireless Communications & Networking*. https://doi.org/10.1016/B978-0-12-373580-5.X5033-9.

Hu, X. (2012). *Wireless Sensor Network : Characteristics and Architectures*. 6(12), 1398–1401.

Ifzarne, S., Hafidi, I., & Idrissi, N. (2020). Compressive Sensing Based on Homomorphic Encryption and Attack Classification using Machine Learning Algorithm in WSN Security. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3386723.3387859.

J. Romberg. (2015). Convex relaxation. *Springer Optimization and Its Applications*, 103, 115–126. https://doi.org/10.1007/978-3-662-46356-7_6.

Jun-Ya Lee, Lin, W.-C., & Huang, Y.-H. (2014). A lightweight authentication protocol for RFID. *Communications in Computer and Information Science*, *448 CCIS*, 110–121. https://doi.org/10.1007/978-3-662-44893-9_10.

Khosravy, M., Nitta, N., Nakamura, K., & Babaguchi, N. (2020). Compressive sensing theoretical foundations in a nutshell. *Compressive Sensing in Healthcare*, 1–24. https://doi.org/10.1016/B978-0-12-821247-9.00006-8.

Kim, H. S., Abdelzaher, T. F., & Kwon, W. H. (2003). Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks. *SenSys'03: Proceedings of the First International Conference on Embedded Networked Sensor Systems*, *January*, 193–204. https://doi.org/10.1145/958491.958515.

Kordafshari, M. S., Pourkabirian, A., Faez, K., & Rahimabadi, A. M. (2009). Energy-efficient SPEED routing protocol for wireless sensor networks. *Proceedings of the 2009 5th Advanced International Conference on Telecommunications, AICT 2009*, *January 2016*, 267–271. https://doi.org/10.1109/AICT.2009.52.

Liu, Y. M., Wu, S. C., & Nian, X. H. (2009). The architecture and characteristics of wireless sensor network. *ICCTD 2009 - 2009 International Conference on Computer Technology and Development*, 1(561), 561–565. https://doi.org/10.1109/ICCTD.2009.44.

Luiz, P., Tavares, A., & RS, J. M. (2016). Reconstruction algorithms in compressive sensing: an overview. *11th Edition of the Doctoral Symposium in Informatics Engineering (DSIE-16)*, *February*, 127–137.

Manchanda, R., & Sharma, K. (2020). A Review of Reconstruction Algorithms in Compressive Sensing. *Proceedings - 2020 International Conference on Advances in Computing, Communication and Materials, ICACCM 2020*, 322–325. https://doi.org/10.1109/ICACCM50413.2020.9212838.

Martinez, J., Mejia, J., & Mederos, B. (2016). Compress sensing for wireless sensor networks using gossip pairwise algorithm and optimisation algorithms. *2016 IEEE Ecuador Technical Chapters Meeting, ETCM 2016*. https://doi.org/10.1109/ETCM.2016.7750847.

Masoum, A., Meratnia, N., & Havinga, P. J. M. (2013). A distributed compressive sensing technique for data gathering in Wireless Sensor Networks. *Procedia Computer Science*, 21, 207–216. https://doi.org/10.1016/j.procs.2013.09.028.

Nasr, S., & Quwaider, M. (2020). LEACH Protocol Enhancement for Increasing WSN Lifetime. *2020 11th International Conference on Information and Communication Systems, ICICS 2020*, 102–107. https://doi.org/10.1109/ICICS49469.2020.239542.

Olakanmi, O. O., & Dada, A. (2018). Wireless Sensor Networks (WSNs): Security and Privacy Issues and Solutions. *Intech*, *WSN*, 18.

Rehmani, M. H., & Pathan, A. S. K. (2016). Emerging communication technologies based on wireless sensor networks: Current research and future applications. In *Emerging Communication Technologies Based on Wireless Sensor Networks: Current Research and Future Applications*.

Shabbir, N., & Hassan, S. R. (2016). Routing Protocols for Wireless Sensor Networks (WSNs). *Intech*, *tourism*, 13.

Sharma, S., & Jena, S. K. (2014). Data Dissemination Protocol for Mobile Sink in Wireless Sensor Networks. *Journal of Computational Engineering*, 2014, 1–10. https://doi.org/10.1155/2014/560675.

Singh, K. (2015). WSN LEACH based protocols: A structural analysis. *2015 International Conference and Workshop on Computing and Communication, IEMCON 2015*. https://doi.org/10.1109/IEMCON.2015.7344478.

Taghouti, M. (2020). Compressed sensing. *Computing in Communication Networks*, 197–215. https://doi.org/10.1016/B978-0-12-820488-7.00023-2.

Yufeng, G., Jiang, Y., Rong, Z., & Kai, X. (2013). *Algorithm Based on Power Communication*. 497–500.